

[首页](#)[官方教程](#)[视频课程](#)[社区开源](#)[正在开发中.....](#)[登录](#)

Spring Data JPA系列：使用@Query注解（Using @Query）

经过几天的折腾，终于到了学习一个重量级的查询方式上，使用@Query注解，使用注解有两种方式，一种是JPQL的SQL语言方式，一种是原生SQL的语言，略有区别，后者我们更熟悉一些。话不多说，看代码。

1、在CustomerRepository里添加

```
/**
 * 模糊匹配
 * @param bauer
 * @return
 */
@Query("select c from Customer c where c.firstName=?1")
Customer findByFirstName2(String bauer);

@Query("select c from Customer c where c.lastName=?1 order by
List<Customer> findByLastName2(String lastName);

/**
 * 一个参数，匹配两个字段
 * @param name2
 * @return
 * 这里Param的值和=:后面的参数匹配，但不需要和方法名对应的参数值对应
 */
@Query("select c from Customer c where c.firstName=:name or c
List<Customer> findByName(@Param("name") String name2);

/**
 * 一个参数，匹配两个字段
 * @param name
 * @return
 * 这里的%只能放在占位的前面，后面不行
 */
@Query("select c from Customer c where c.firstName like %?1")
List<Customer> findByName2(@Param("name") String name);

/**
 * 一个参数，匹配两个字段
 * @param name
 * @return
 * 开启nativeQuery=true，在value里可以用原生SQL语句完成查询
 */
@Query(nativeQuery = true,value = "select * from Customer c w
List<Customer> findByName3(@Param("name") String name);
```

2、在CustomerController内添加

```
/**
 * @Query注解方式查询
 * 查询FirstName为指定字符串
 */
@RequestMapping("/findByFirstName2")
public void findByFirstName2(){
    Customer customer = repository.findByFirstName2("Bauer");
    if(customer!=null){
        System.out.println(customer.toString());
    }
}
```

[Spring Data](#)

阅读	评论	喜欢
319	0	0

[喜欢](#)[SpringForAll](#)

文章	问答	粉丝
79	6	236

[关注](#)

```

    }
    System.out.println("-----")
}

/**
 * @Query注解方式查询
 * 查询LastName为指定字符串
 */
@RequestMapping("/findByLastName2")
public void findByLastName2(){
    List<Customer> result = repository.findByLastName2(
        for (Customer customer:result){
            System.out.println(customer.toString());
        }
    System.out.println("-----")
}

/**
 * @Query注解方式查询,
 * 用@Param指定参数, 匹配firstName和lastName
 */
@RequestMapping("/findByName")
public void findByName(){
    List<Customer> result = repository.findByName("Baue
    for (Customer customer:result){
        System.out.println(customer.toString());
    }
    System.out.println("-----")
}

/**
 * @Query注解方式查询,使用关键词like
 * 用@Param指定参数, firstName的结尾为e的字符串
 */
@RequestMapping("/findByName2")
public void findByName2(){
    List<Customer> result = repository.findByName2("e")
    for (Customer customer:result){
        System.out.println(customer.toString());
    }
    System.out.println("-----")
}

/**
 * @Query注解方式查询, 模糊匹配关键字e
 */
@RequestMapping("/findByName3")
public void findByName3(){
    List<Customer> result = repository.findByName3("e")
    for (Customer customer:result){
        System.out.println(customer.toString());
    }
    System.out.println("-----")
}

```

可能看了上面的代码有些疑惑, 这里做一下解释:

? 加数字表示占位符, ? 1代表在方法参数里的第一个参数, 区别于其他的index, 这里从1开始

=: 加上变量名, 这里是与方法参数中有@Param的值匹配的, 而不是与实际参数匹配的

JSQL的语法中, 表名的位置对应Entity的名称, 字段对应Entity的属性, 详细语法见相关文档

